

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Ivo Voika

Võimatu geomeetria sõlmepõhises maailmas

Bakalaureusetöö (6 EAP)

Juhendaja: Raimond-Hendrik Tunnel, MSc

Tartu 2017

Võimatu geomeetria sõlmepõhises maailmas

Lühikokkuvõte:

Käesolev bakalaureusetöö annab ülevaate sõlmepõhisest maailmast ning kuidas selles kujutada võimatut geomeetriat. Kirjeldatakse sõlmepõhise maailma eeliseid võrreldes tavalise lähenemisega ja tuuakse välja töö käigus tekkinud probleeme. Töö käigus valmis programm, kus on implementeeritud sõlmepõhine maailm. Programmis on toodud näiteid kolme erinevat tüüpi võimatust geomeetriast.

Võtmesõnad:

Sõlmepõhine maailm, võimatu geomeetria, arvutigraafika, arvutimängud

CERCS:

P170: Arvutiteadus, arvanalüüs, süsteemid, juhtimine;

P175: Informaatika, süsteemiteooria.

Impossible Geometry in a Node-Based World

Abstract:

This Bachelor's thesis gives an overview of node-based world and how to represent impossible geometry within it. It describes the advantages compared to the usual approach and brings out problems that arose. The result of this work is a program that implements node-based world. The program has examples of three types of impossible geometry.

Keywords:

Node-based world, impossible geometry, computer graphics, computer games

CERCS:

P170: Computer science, numerical analysis, systems, control;

P175: Informatics, systems theory.

Sisukord

1.	Sissejuhatus	5
2.	Võimatu geomeetria	6
3.	Sõlmepõhine maailm.....	9
3.1	Võrdlus tavalise lähenemisega	11
3.1.1	Ujukomaarvu täpsus.....	11
3.1.2	Sihtmärgile osutamine.....	12
3.1.3	Vibunooole probleem.....	13
3.2	Minu implementatsioon.....	13
3.2.1	Kasutatud teegid ja tehnoloogiad	14
3.2.2	Sõlmede koostamine	14
3.2.3	Sõlmede ühendused.....	15
3.2.4	Sõlmede ja ühenduste implementatsioon	16
3.3	Esinenud probleemid ja nende lahendused.....	16
3.3.1	Ebavajalike sõlmede eemaldamise varjamine.....	16
3.3.2	Ühe- ja kahepoolsed ühendused.....	18
4.	Kasutusnäited	20
4.1	Otseteed geomeetria vahel.....	20
4.1.1	Illusiooni olemus.....	20
4.1.2	Kasutus	21
4.1.3	Implementatsioon.....	22
4.2	Lõpmatu koridor	23
4.2.1	Illusiooni olemus.....	23
4.2.2	Kasutus	23
4.2.3	Implementatsioon.....	24
4.3	Ruumiliselt kattuvate aladega geomeetria.....	26

4.3.1	Illusiooni olemus	27
4.3.2	Kasutus	28
4.3.3	Implementatsioon	30
5.	Kokkuvõte	31
6.	Viidatud kirjandus	32
	Lisad	34
I.	Programm	34
II.	Programmi kasutusõpetus	35
III.	Videod	36
IV.	Litsents	37

1. Sissejuhatus

Võimatut geometriat saab kasutada arvutimängudes huvitava visuaalse efekti loomiseks. Kujundeid, nagu näiteks lõputuid treppe ja Penrose'i kolmnurka [1], on kasutatud sellistes mängudes nagu Monument Valley¹ ja Antichamber². Samas on saadaval vähe informatsiooni, kuidas implementeerida sellist geometriat virtuaalsetes mängumaailmates nõnda, et mängumaailma geometria oleks võimatu geometria. Üks võimalik lahendus oleks sõlmepõhine lähenemine, mida on kasutatud mängus Dungeon Siege (2002)³. Seda lahendusvarianti on kirjeldatud tolle mängu autorite poolt artiklis “The Continuous World of Dungeon Siege” [2].

Käesolev töö uurib ja kirjeldab sõlmepõhist lähenemist, kus virtuaalmaailma geometria jagatakse omavahel ühendatud osadeks ehk sõlmedeks. Sõlmi kuvatakse dünaamiliselt vastavalt vajadusele. Üheaegselt kuvatud sõlmed peavad moodustama alati võimaliku geometria, et anda mängijale kujutlus terviklikust maailmast. Teatud sõlmede korraga kuvamine tekitab võimatu geometria, mida ei saa terviklikult kujutada. Näiteks 8-paigutusega ruumis tuleks joonistada teatud sõlmed teineteise peale, selliseid sõlmi ei tohiks üheaegselt kuvada (täpsemalt käsitletud peatükis 4.3).

Töö raames valmis arvutiprogramm (lisa I), kus on implementeeritud sõlmepõhine maailm ning mõned näidised süsteemi võimalike kasutuste kohta. Valminud programm annab võimaluse kogeda töös kirjeldatud võimatuid geometriaid parema arusaamise eesmärgil. Samuti on võimalik kasutada programmi eeskujuna sõlmepõhise maailma implementeerimisel.

Peatükk 2 annab lühikese ülevaate võimatu geometria tekkest ning kasutusest kunstis. Peatükis 3 on toodud välja põhilised erinevused sõlmepõhise ja tavalise lähenemise vahel. Lisaks on peatükis 3 kirjeldatud antud töö käigus valminud programmis implementeeritud sõlmepõhise maailma tööpõhimõtted ning süsteemi loomisel tekkinud probleemid. Peatükis 4 on kirjeldatud kolm erinevat võimalikku kasutust antud süsteemile koos kirjelduste ja näidetega.

Lisades on kaasas programmi käivitamise õpetus ja nõuded arvutile (lisa I), kasutusõpetus (lisa II) ning videod rakenduses implementeeritud näidistest (lisa III).

¹ Monument Valley, www.monumentvalleygame.com (10.05.2017)

² Antichamber, www.antichamber-game.com (10.05.2017)

³ Dungeon Siege – Wikipedia, https://en.wikipedia.org/wiki/Dungeon_Siege (10.05.2017)

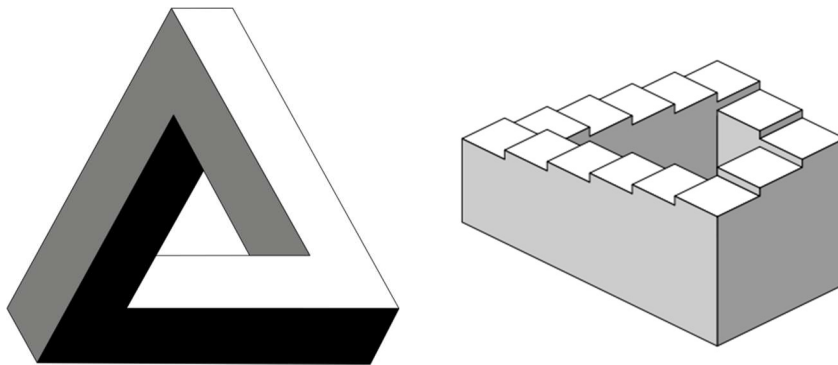
2. Võimatu geomeetria

Optiliste illusioonide kohta on öeldud, et:

Optiline illusioon on illusioon, mille põhjustab visuaalsüsteem ja mida iseloomustavad visuaalselt tajutavad kujutised, mis erinevad objektiivsest reaalsusest. Silma kogutud informatsioon on töödeldud ajus, et anda ettekujutus, mis ei ühti stiimuli füüsiliste mõõtmetega (tõlgitud ing. k.) [3].

Võimatud objektid moodustavad optiliste illusioonide alamklassi, milles 2D joonistused kujutavad näiliselt objekte, mis ei saaks eksisteerida 3D ruumis [4]. Võimatute objektide häiriva olemuse põhjustab meie tahe kujutada 2D joonistust kui kolmemõõtmelist objekti [5].

Oscar Reutersvärdit loetakse võimatute kujundite esiisaks [6:261]. Ta lõi võimatu kolmnurga (illustratsioon 1, vasakul) aastal 1934, olles kõigest 19-aastane, ja võimatud trepid (illustratsioon 1, paremal) aastal 1937. Sellel ajal peeti tema teoseid kõigest veidrusteks [6:262]. Aastal 1958 näitas Reutersvärdi sõber talle Lionel ja Roger Penrose'i artiklit võimatutest objektidest. Lionel ja Roger Penrose olid iseseisvalt loonud võimatu kolmnurga ja võimatud trepid. Tänapäevalgi tuntakse neid kujundeid Penrose'i trepi ja Penrose'i kolmnurgana.

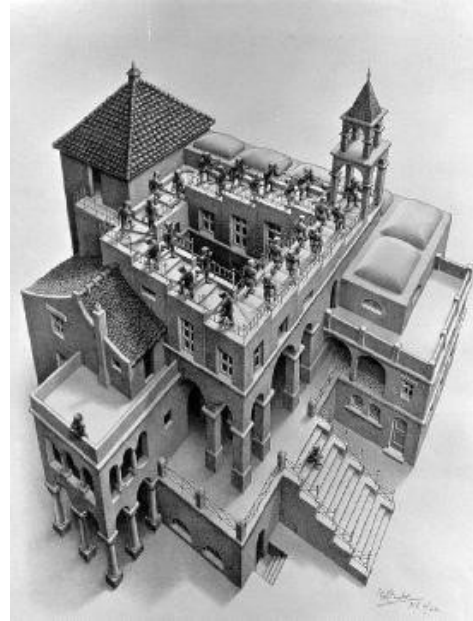


Illustratsioon 1. Penrose'i kolmnurk (vasakul)⁴ ja Penrose'i trepid (paremal)⁵.

Kunstnik Maurits Cornelis Escher sai Penrose töödest inspiratsiooni ning valmistas mitmeid erinevaid teoseid, milles kasutas võimatut geomeetriat, eg. “Kosk” (illustratsioon 2, vasakul) ja “Trepist üles ja alla” (illustratsioon 2, paremal).

⁴ Penrose'i kolmnurk, <https://commons.wikimedia.org/wiki/File:Penrose-dreieck.svg> (10.05.2017)

⁵ Penrose'i trepid, https://commons.wikimedia.org/wiki/File:Impossible_staircase.svg (10.05.2017)



Illustratsioon 2. M. C. Escher teosed “Kosk” (vasakul)⁶ ja “Trepist üles ja alla” (paremal)⁷.

“Kosk” on kuulus litograafiline trükk, milles on esindatud paradoksaalsed elemendid. Vesi kose jalamil näib voolavat allamäge kuni jõuab tagasi kose tippu [7]. Teoses “Trepist üles ja alla” kasutas Escher inspiratsioonina Penrose’i treppe ning lõi lõputud trepid, mida mööda grupp inimesi lõputult üles ja alla liiguvad [8].

Võimatute geomeetria loomiseks on optilisi illusioone kasutatud ka arvutimängudes Echochrome⁸ ja Monument Valley⁹.

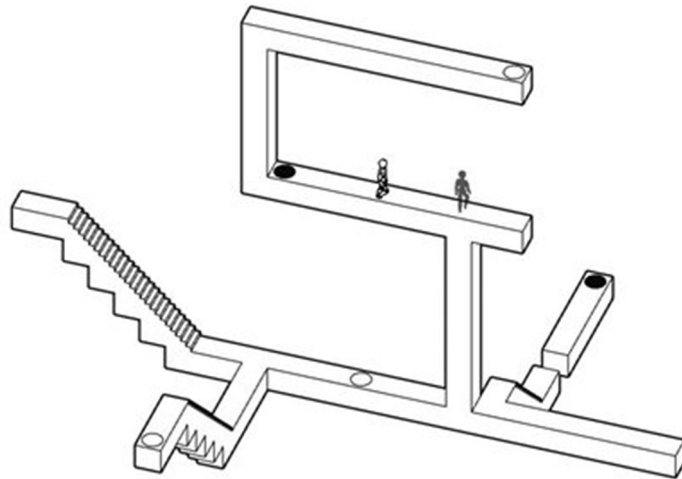
Mängus Echochrome (illustratsioon 3) juhib mängija kaamera perspektiivi samas kui mängu tegelane liigub iseseisvalt. Perspektiivi muutmisega peab mängija näiteks moodustama võimalikke geomeetriaid või katma kinni auke geomeetrias, et avada tegelasele uusi teid ning seeläbi juhatada tegelane taseme lõppu.

⁶ M. C. Escher – „Kosk“, https://en.wikipedia.org/wiki/File:Escher_Waterfall.jpg (10.05.2017)

⁷ M. C. Escher – „Trepist üles ja alla“, https://en.wikipedia.org/wiki/File:Ascending_and_Descending.jpg (10.05.2017)

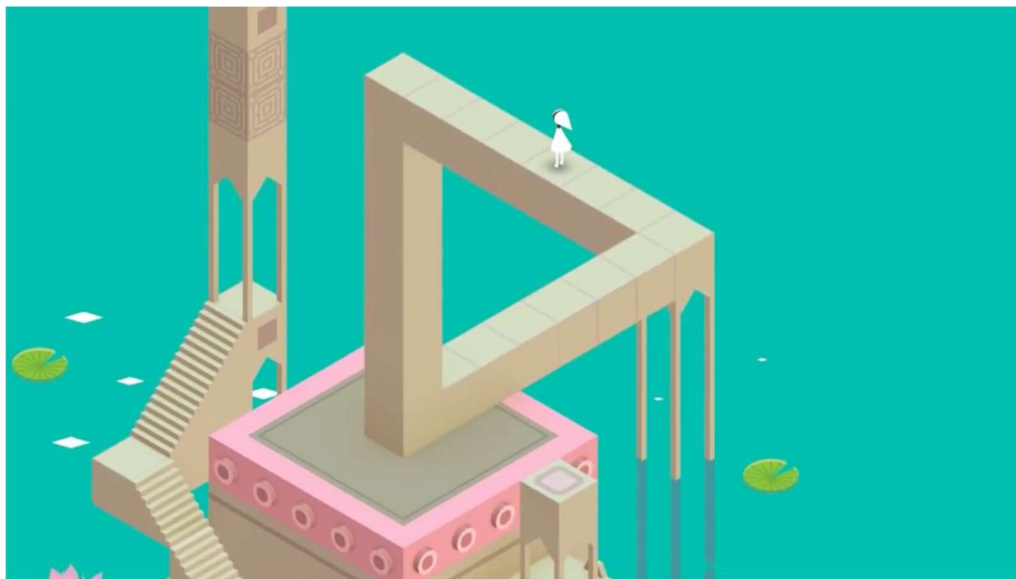
⁸ Echochrome – Wikipedia, <https://en.wikipedia.org/wiki/Echochrome> (10.05.2017)

⁹ Monument Valley, www.monumentvalleygame.com (10.05.2017)



Illustratsioon 3. Tase mängust Echochrome¹⁰.

Mängus Monument Valley (illustratsioon 4) peab mängija tegelase läbi taseme juhtima, et edasi järgmisesse tasemesse saada. Selleks saab mängija pöörata ja liigutada tasemes olevaid geometriaid, et eesmärk saavutada. Üheks osaks ülesannete lahendamisel on samuti võimatute geometriate moodustamine ja läbimine [9].



Illustratsioon 4. Mängus Monument Valley kujutatud Penrose'i kolmnurk¹¹.

Käesolevas töös vaatleme edaspidi optilisi illusioone arvutimängude kontekstis ja uurime, kuidas neid esindada ning millised on nende võimalikud kasutused.

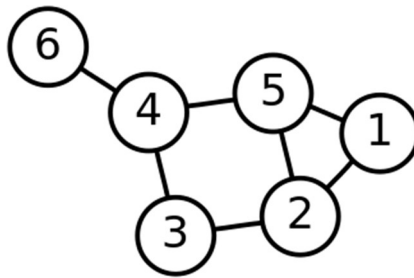
¹⁰ Echochrome – Educational Game Review, <http://learningworksforkids.com/playbooks/echochrome/> (10.05.2017)

¹¹ Monument Valley: El ciao bella de Ustwo, <http://www.startvideojuegos.com/monument-valley-el-ciao-bella-de-ustwo> (10.05.2017)

3. Sõlmepõhine maailm

Artiklis “The Continuous World of Dungeon Siege” [2] on kirjeldatud sõlmepõhise süsteemi kasutust 2002. aasta mängus Dungeon Siege. Algselt oli süsteemi eesmärk võimaldada mängutegelasel rännata läbi kogu mängumaailma ilma laadimisekraanideta. Lisaks aga võimaldas see süsteem neil luua maailma, mis ei pidanud järgima Eukleidilise ruumimudeli reegleid [2:16].

Sõlmepõhine virtuaalmaailm on sidus graaf, mille iga sõlm esindab osa maailma geomeetriast. Graaf on matemaatiline objekt, mis koosneb mingist hulgast tippudest ja neid tippe paarikaupa ühendavatest servadest [10] (illustratsioon 5). E. G. Bravo ja J. A. García on samuti kasutanud graafe, sarnaselt siinkäsitletule, võimatute geomeetria kujutamiseks oma artiklis „Yturalde: Impossible Figure Generator“ [11].

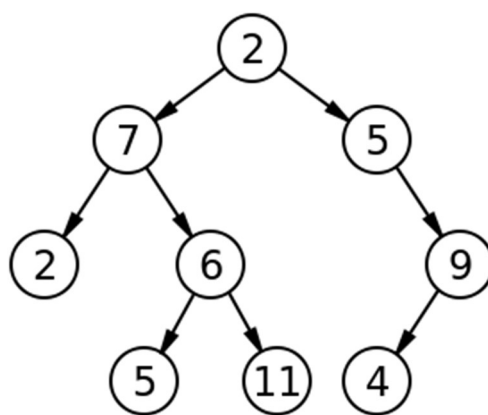


Illustratsioon 5. Näide graafist. Tipud on tähistatud ringidega ja servad tippude vahele tõmmatud joontega¹².

Kui kahte graafi tippe ühendaval serval ei ole suunda määratud, siis nimetatakse seda orienteerimata ehk suunamata servaks ehk lühemalt lihtsalt servaks. Kui aga serval on suund määratud, nimetatakse seda orienteeritud ehk suunatud servaks ehk kaareks. Graafi nimetatakse sidusaks, kui iga kahe tipu korral leidub neid tippe ühendav ahel [10].

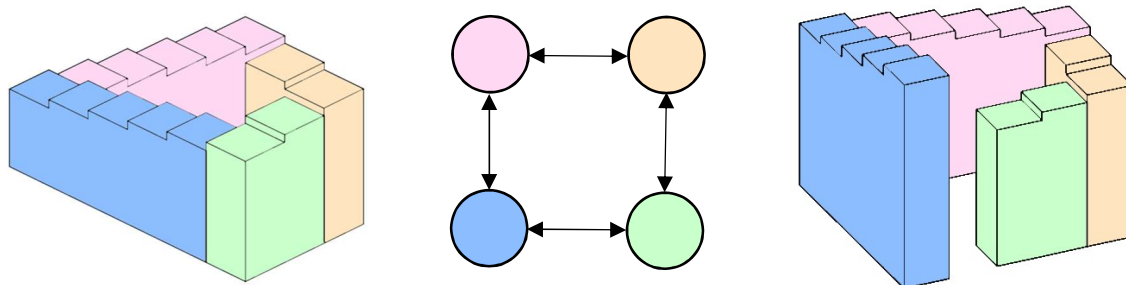
Tavaliselt nimetatakse sõlmeks puu tippu, millel on mõni alluv. Puu on sidus tsükliteta graaf [10] (illustratsioon 6). Artiklis “The Continuous World of Dungeon Siege” nimetatakse maailmaosi inglise keeles sõnaga *node* [2], mille tõlge eesti keeles on “sõlm”. Seega nimetatakse antud töös maailmaosi sõlmedeks.

¹² Graaf, <https://commons.wikimedia.org/wiki/File:6n-graf.svg> (10.05.2017)



Illustratsioon 6. Näide puustruktuurist - kahendpuu¹³.

Penrose'i treppe kolmemõõtmelises ruumis esineda ei saa. Küll aga saaks jätta illusiooni sellisest geomeetriast. Esiteks jagada Penrose'i trepi geomeetria neljaks erinevaks osaks (joonis 7, vasakul). Iga osa saab esindada graafis sõlmene (joonis 7, keskel). Jättes eelneval graafil ühe vabalt valitud serva ära, saab võimaliku geomeetria (joonis 7, paremal). Ära võetud ühendus moodustab katkendi. Virtuaalmaailmas saaks katkendi varjamiseks tekitada see väljapoole vaatevälja. Vastavalt vaatevälja liikumisele oleks võimalik katkendit geomeetrias dünaamiliselt liigutada nõnda, et see poleks vaateväljas näha.



Joonis 7. Vasakul – Penrose'i trepp illustratsioonilt 1 (paremal) jagatud värvidega neljaks osaks. Keskul – Penrose'i trepp graafi kujul. Paremal – katkendiga trepp.

Kuigi Penrose'i trepid kolmemõõtmelises ruumis esineda ei saa, on Nakatsu K, Takahashi T ja Moriya T suutnud luua viisi kolmemõõtmelises ruumis Penrose'i trepi interaktiivselt pööramiseks [12].

Edasi vaatleme, millised on sõlmepõhise maailma kõige olulisemad erinevused võrreldes tavalise lähenemisega.

¹³ Kahendpuu, https://commons.wikimedia.org/wiki/File:Binary_tree.svg (10.05.2017)

3.1 Võrdlus tavalise lähenemisega

Antud peatükis on kirjeldatud ja täpsemalt seletatud mõned suuremad erinevused tavalise lähenemisega, mis on artiklis “The Continuous World of Dungeon Siege” [2] välja toodud.

3.1.1 Ujukomaarvu täpsus

Mängumaailmas olevate objektide positsioone määratakse koordinaatidega kahe- või kolmemõõtmelises Eukleidilises ruumis. Arvutiprogrammides talletatakse neid koordinaate üldiselt lihttäpsusega ujukomaarvudena (*float* andmetüüp). Tüüpiliselt teisendatakse arvuti-mängudes Eukleidilise ruumi üks ühik kas meetriks või sentimeetriks. Seega koordinaadid tähistavad üldiselt meetreid või sentimeetreid koordinaatide alguspunktist.

Java keeles on ujukomaarvu kuju

$$(-1)^s \cdot m \cdot 2^{(e-127)}$$

ja iga ujukomaarvu hoidmiseks kasutatakse 32-te bitti [13]:

- Märki bitt s (bitt 32). 1 – negatiivne, 0 – positiivne.
- Eksponent e (bitid 24-31).
- Arvu tüvi m (bitid 1-23).

Sedasi on võimalik esindada piiratud hulk erinevaid reaalarve. Tegelik reaalarvude hulk on aga lõpmatu. See tähendab, et tuleb teha ja kasutada lähendeid. Mida suurem on arvu tühi, seda vähemaks jääb komakohtade täpsus. Näiteks:

- Arvule $\frac{1}{3}$ vastab ujukomaarv 0,3333333432674408.
- Arvule 100000,333 vastab ujukomaarv 100000,3359375. Komakohti on juba märgatavalt vähemaks jäänud võrreldes eelneva näidisega.
- Arvudele 123456,122 ja 123456,128 vastab ujukomaarv 123456,125. Arvuti jaoks on sisuliselt tegu samade arvudega.

Kui mänguobjektid asuvad koordinaatteljestiku nullkohast väga kaugel, siis võivad nad ettearvamatult käituda olukordades, kus on olulised täpsed arvutused. Näiteks kokkupõrkekontrollis või objektide joonistamisel ekraanile [2:3]. Seepärast on oluline, et objektid ei asuks koordinaatteljestiku alguskohast liialt kaugel.

Topelttäpsusega ujukomaarvude (*double* andmetüüp) kasutamine annaks täpsust juurde, aga tavakasutajatele mõeldud graafikakaardid on optimeeritud lihttäpsusega ujukomaarvude arvutusteks. Topelttäpsusega ujukomaarvude jõudlus moodsatel tavakasutajatele mõeldud

nVidia GeForce seeria graafikakaartidel, näiteks GTX 1080, on vaid 1/32 lihttäpsusega ujukomaarvude jõudlusest [14]. Seega poleks mõistlik kasutada programmis topelttäpsusega ujukomaarve, kuna saab vaid 1/32 lihttäpsusega ujukomaarvude jõudlusest.

Tavaliselt saadakse suurte ujukomaarvude täpsuse probleemist üle jagades mängumaaailm erinevateks tasemeteks, millel igal on oma koordinaatide alguspunkt. Kui mängija jõuab kohta, kus on vaja uus tase laadida, siis jääb mäng seisma ja kuvab ekraanile laadimise teadet seniks kuni uus tase on mällu laetud.

Mängus Dungeon Siege planeeriti luua suurt mängumaaailma. Lisaks sooviti, et see maailm oleks läbitav ilma laadimisekraanideta. Tavalise lähenemise puhul oleks ujukomaarvu täpsuse kadu probleemiks olnud. Nende lahenduseks oli jagada mängumaaailm väikesteks osadeks ehk sõlmedeks, tüüpiliselt suurusega $4 \cdot 4m$ või $4 \cdot 8m$, mida sai kiiresti mällu laadida ilma laadimisekraani kuvamata. Kui kasutaja liikus mängumaaailmas uude sõlme, liigutati maailma koordinaatteljestiku alguspunkt antud sõlme raamidesse. Seega polnud tegelane kunagi kaugel koordinaatteljestiku alguspunktist.

3.1.2 Sihtmärgile osutamine

Sihtmärgile osutamist kasutatakse tihti mängija juhtimiseks õigesse kohta. Visuaalselt näeb lahendus üldiselt välja kui nool, mis osutab mängija sihtkoha poole (illustatsioon 7).



Illustatsioon 7. Mängus Crazy Taxi osutas roheline nool mängija sihtmärgile¹⁴.

Sõlmepõhises maailmas aga pole ühtset nullpunkti, mille suhtes saaks üle kõigi sõlmede positsiooni üheselt määrata. Lisaks võivad sõlmed ruumiliselt kattuda (lähemalt peatükis

¹⁴ Crazy Taxi, https://en.wikipedia.org/wiki/File:Crazy_taxi Ride.jpg (10.05.2017)

4.3), mistõttu oleks mängijal raske eristada, millisele sõlmele nool osutab. Võimalik lahendus oleks leida lühim teekond graafis sihtmärgini ning osutada noolega teekonna järgmise sõlme poole.

3.1.3 Vibunool probleem

Artiklis “The Continuous World of Dungeon Siege” [2] on kirjeldatud probleem vibunooltega. Kui mängija tegelane lasi vibuga noole lendu nii, et selle trajektoor läbis mitu sõlme, oli vaja igal ajahetkel kokkupõrgete testimiseks teada, millises sõlmes nool parajasti asus. Mängus Dungeon Siege kasutati kiirtejälitust (ing. k. *ray tracing*) ja rekursiivset sõlmede graafi läbimist, et leida hetkel sobivaim sõlm, kus nool potentsiaalselt asub. Otsingul oli kontrollitavate sõlmede arv piiratud. Näiteks lastes noole üle kuristiku, kus ühel ja teisel pool kuristikku asuvad sõlmede polnud omavahel otse ühendatud, võis juhtuda, et õiget sõlme ei leitud sobiva ajaga ning otsing katkestati. Kui otsing ebaõnnestus, siis ei tehtud noole ja sihtmärgi vahel kokkupõrketesti ning nool lendas läbi sihtmärgi.

3.2 Minu implementatsioon

Dungeon Siege süsteemis laeti mällu kõik sõlmed, mis olid hetkel ekraanil olevate sõlmede otsesed naabrid ning jäid kaamera vaatealasse. Minu lahenduses laetakse taseme kõik sõlmed kohe mällu ning kuvatakse hetkel aktiivne sõlm ning selle kõik vahetud naabersõlmed. Aktiivseks loetakse sõlm, milles kasutaja tegelane parajasti asub. Minu lahenduse pole tehtud dünaamilist sõlmede mällu laadimist ning nende sealt eemaldamist, sest tasemed pole suured ning seega ei teki jõudlusprobleeme kui ka kõik sõlmed on korraga mällu laetud. Lisaks polnud see funktsionaalsus oluline sõlmepõhise maailma demonstreerimiseks. See funktsionaalsus võib olla võimalik tulevane edasiarendus antud programmi jaoks.

Mängus Dungeon Siege pidid kõik parajasti vaatealas asuvad sõlmed moodustama võimaliku geomeetria. Minu süsteemis, kus kuvatakse ainult aktiivne sõlm ja selle naabrid, peavad ainult kuvatud sõlmed võimaliku geomeetria moodustama. See võimaldas kompaktsemat tasemete disaini võrreldes Dungeon Siege süsteemiga.

Käesolevas töös valminud rakendus kujutab 2D virtuaalmaailma ülaltvaates. See seadis piirangud võimalikele kujutatavatele võimatutele objektidele. Näiteks Penrose'i trepid ei oma erilist efekti, kuna puudub võimalus kõrguses üles ja alla liikuda.

3.2.1 Kasutatud teegid ja tehnoloogiad

Kasutatud programmeerimiskeeleks oli Java 8¹⁵, sest autoril oli kõige rohkem kogemust selle keelega. Kuigi tihti eelistatakse graafikaprogrammide loomiseks jõudlusele orienteeritud keeli nagu näiteks C++, siis Java kasuks räägib lihtsam mäluhaldus ja samas piisavalt hea jõudlus, et saada reaaliajase hakkama antud ülesandega. Elementaarse katse käigus mõõdetud keskmiseks kaadrisageduseks valminud programmis oli 1100 kaadrit sekundis. Katse viidi läbi arvutil, millel oli protsessor i7-3770K ja graafikakaart GeForce GTX 960.

Lihtsustamaks suhtlust OpenGL ja Java vahel, kasutasin teeki Lightweight Java Game Library (LWJGL)¹⁶. OpenGL¹⁷ on standardiseeritud liides (API, ingl. k. *application programming interface*) graafikakaardiga suhtlemiseks. LWJGL sisaldab ka utiliiti GLFW¹⁸, mis lihtsustab programmiakende haldamist ja kasutaja sisendi käitlemist. Lisaks on kasutusel LWJGL poolt soovitatud matemaatika-teek JOML¹⁹ (Java OpenGL Math Library) [15]. Maailmaosade koostamiseks oli kasutusel programm Tiled²⁰, mis väljastab *.tmx* laiendiga XML-formaadis faile, mille käitlemiseks käesoleva töö raames valminud programmis kasutatakse libtiled-java²¹ teeki. Lähtekoodi ehitamiseks oli kasutusel Gradle²² tehnoloogia.

Alternatiivsete Java raamistike (näiteks LibGDX) või mängumootorite puhul on mugavama kasutamise jaoks tavaliselt paika pandud standardne viis mängu ülesehituseks. Näiteks võivad olla paika pandud süsteemid kaamerate halduseks või maailma esitluseks. Raamistike kasutusviisidest erinevalt asjade tegemiseks peaks väga hästi tundma nende vahendite võimalusi. Projekti alustades oli aga eeldus, et ainult standardsetest kasutusviisidest ei piisaks.

3.2.2 Sõlmede koostamine

Antud töös kasutatud sõlmed on riskülikukujulised ning koostati programmiga Tiled, millega saab lihtsalt valmistada paanidest (ingl. k. *tile*) koosnevaid 2D mängumaailmu. Tuleb vaid importida paanide komplekt (ingl. k. *tileset*), mida soovitakse mängumaailmas kasutada (illustratsioon 9, (C)), valida mängumaailma suurus ning täita ruudustik (illustratsioon 9, (A)). Kõik koostatud sõlmed koosnesid kahest andmekihist. Alumine kiht moodustas sõlme

¹⁵ Java, <https://www.java.com> (10.05.2017)

¹⁶ LWJGL – Lightweight Java Game Library, <https://www.lwjgl.org> (10.05.2017)

¹⁷ OpenGL, <https://www.opengl.org> (10.05.2017)

¹⁸ GLFW, <http://www.glfw.org> (10.05.2017)

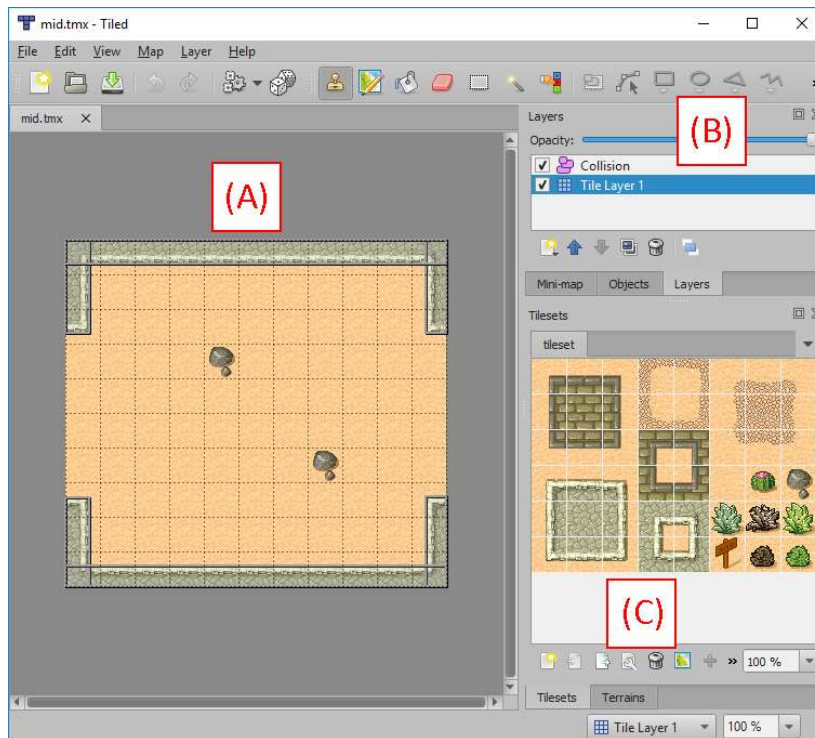
¹⁹ JOML – Java OpenGL Math Library, <https://github.com/JOML-CI/JOML> (10.05.2017)

²⁰ Tiled Map Editor, <http://www.mapeditor.org> (10.05.2017)

²¹ libtiled Java, <https://github.com/bjorn/tiled/tree/master/util/java/libtiled-java> (10.05.2017)

²² Gradle Build Tool, <https://gradle.org> (10.05.2017)

visuaalse poole ja pealmine kiht sisaldas informatsiooni põrketuvastuse jaoks (illustratsioon 9, (B)).



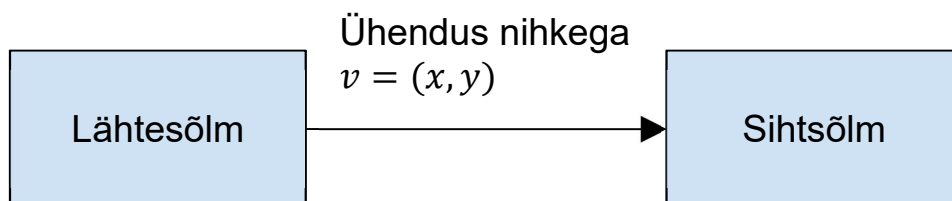
Illustratsioon 9. (A) – mängumaailma vaade. (B) – andmekihid. (C) – imporditud paanide komplektid.

Järgnevalt on kirjeldatud töö raames valminud programmi implementatsioon.

3.2.3 Sõlmede ühendused

Ühendus kirjeldab, millised kaks sõlme on omavahel naabrid, kumb sõlmedest on ühenduse lähtesõlm ning kumb on sihtsõlm. Samuti on ühenduses kirjeldatud nihe sõlmede vahel (joonis 10). Nihe on vajalik, et programm teaks kuhu tuleks naabersõlm hetkel aktiivse sõlme suhtes kuvada. Nihe on kirjeldatud vektoriga

$$v = (x, y), \text{ kus } x, y \in \mathbb{R}.$$



Joonis 10. Joonisel on noolega kujutatud ühendus lähte- ja sihtsõlme vahel.

Programmi jaoks on vajalik teada, kas aktiivne sõlm on ühenduses lähte- või sihtsõlm. Kui aktiivne sõlm on ühenduses lähtesõlmeks, tuleb ühenduse sihtsõlm kuvada nihkega v . Kui aktiivne sõlm on ühenduses sihtsõlmeks, tuleb ühenduse lähtesõlm kuvada nihkega $-v$.

Näide: ühendus kahe sõlme vahel on nihkega $(100, 0)$. Järelikult asub lähtesõlm sihtsõlmest 100 pikslit paremal. Kui aktiivseks sõlmeks on lähtesõlm, kuvatakse sihtsõlm aktiivsest sõlmest paremal, nihkega $(100, 0)$. Kui mängija liigub tegelasega sihtsõlme ning see muudetakse aktiivseks sõlmeks, kuvatakse ühenduse lähtesõlm aktiivsest sõlmest vasakul, nihkega $(-100, 0)$.

Ühendused võivad olla ühe- või kahepoolsed. Seni on kirjeldatud kahepoolseid ühendusi. Ühepoolsetest ühendustest lähemalt peatükis 3.3.2.

Ühenduse loomisel kahe sõlme vahele talletatakse info ühenduse kohta mõlemas sõlmes (k.a ühepoolsete ühenduste puhul). Sõlmede kuvamisel itereeritakse üle kõigi aktiivse sõlme ühenduste kaks korda. Esimesel korral kontrollitakse talletatud ühenduste lähtesõlmi. Kui lähtesõlm pole aktiivse sõlmega sama, siis ühenduse sihtsõlm kuvatakse. Teisel korral kontrollitakse talletatud ühenduste lähtesõlmi. Kui lähtesõlm pole aktiivse sõlmega sama ning ühendus pole ühepoolne, siis ühenduse lähtesõlm kuvatakse.

3.2.4 Sõlmede ja ühenduste implementatsioon

Igat mängumaaailma sõlme esindab objekt `WorldNode`. Seal hoitakse tekstuure, mänguobjekte (nagu näiteks kasutaja tegelane), pörketuvastuse geomeetriat ja sõlme sissetulevaid ning väljuvaid ühendusi teiste sõlmedega. Lisaks hoitakse seal ka `libtiled-java` teegiga programmi laetud `Tiled` kaardifaili objekti.

Sõlmede ühendused on kirjeldatud `NodeLink` objektidega. Selles objektis on tähistatud ühenduse lähte- ja sihtsõlm, märge selle kohta, kas ühendus on ühe- või kahepoolne, ning nihe sõlmede vahel.

3.3 Esinenud probleemid ja nende lahendused

Arenduse käigus tekkis mitu probleemi, millest märkimisväärsemaid on järgnevalt kirjeldatud.

3.3.1 Ebavajalike sõlmede eemaldamise varjamine

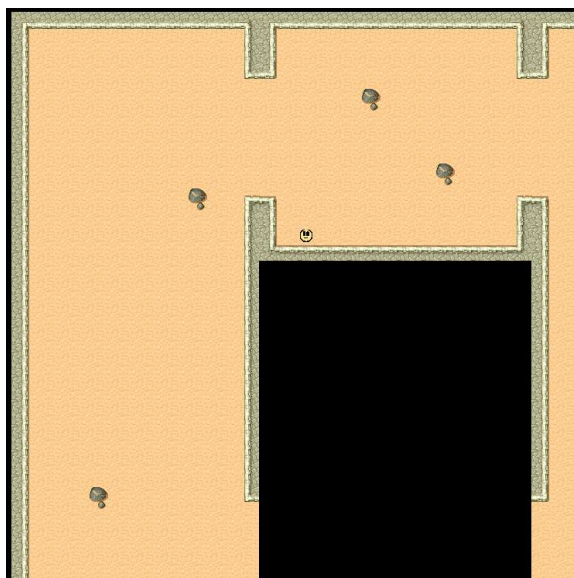
Erinevalt mängus `Dungeon Siege` implementeeritud süsteemist ei laeta minu implementatsioonis sõlmi kaamera vaatevälja põhjal, vaid taseme laadimisel laetakse kõik sõlmed mällu

ning sõlmi kuvatakse aktiivse sõlme põhjal. Seega kui mängija liigub ühest sõlmest teise, tuleb kuvada uue aktiivse sõlme naabrid ning eelnevaid naabreid mitte. Kui mängija näeb, kuidas osad maailmast lihtsalt kaovad, siis võib illusioon terviklikust mängumaailmast kaduda (illustratsioon 11).

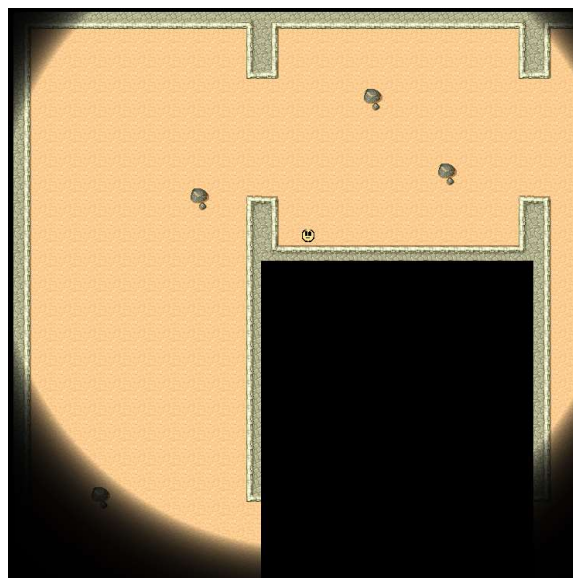
Võimalikuks lahenduseks oleks piirata mängija vaatevälja. Selleks oleks vähemalt kaks viisi:

1. Tekitada ümar valgustatud sõõr ümber tegelase ning kõik muu on varjatud (illustratsioon 13).
2. Joonistada taseme geomeetria põhjal nägemisala (illustratsioon 14).

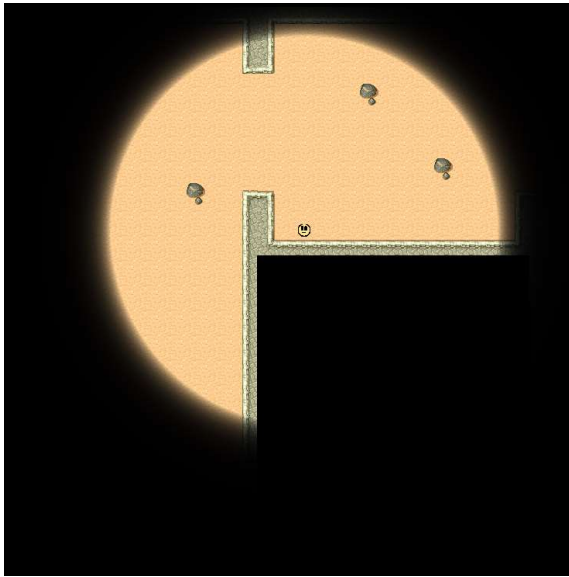
Mõlema lahendusel on sarnased probleemid. Sõlmede kadumised ja tekkimised võivad piisavalt suure valgustatud sõõri puhul siiski näha jääda (illustratsioon 12), mida võimendab ka fakt, et ka teisele poole seinu ja muid takistusi jäävad alad on nähtaval. Selle vältimiseks peaks tasemete disainimisel arvestama valguse sõõri suurusega. Sarnane probleem võib esineda ka teise lahenduse puhul kui eemaldatav sõlm jääb nägemisalasse. Seda saaks aga lahendada vajalikesse kohtadesse paigutatud seinte või muude takistustega, mis piiraksid nägemisala. Võimalus oleks ka mõlemad variandid kombineerida.



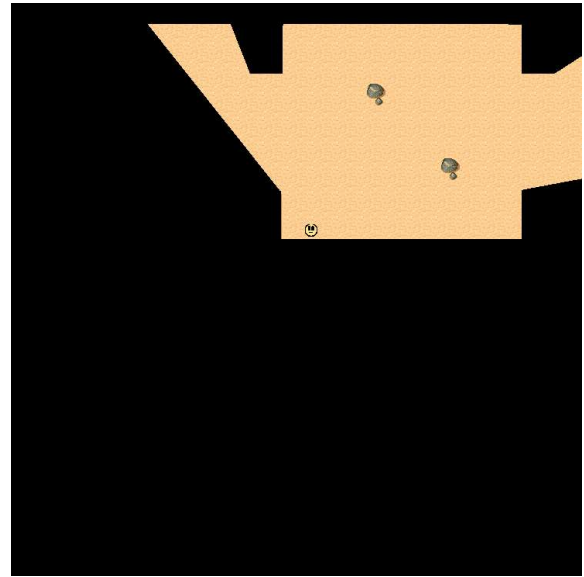
Illustratsioon 11. Ilma varjamiseta. Katkendid geomeetrias on pildil allosas näha.



Illustratsioon 12. Liiga suure valgussõõriga - katkendid geomeetrias on pildi allosas näha.



Illustratsioon 13. Piisavalt väikese valgus-
sõõriga, et varjata katkendid geomeetrias.



Illustratsioon 14. Joonis sellest, kuidas võiks
visuaalselt välja näha nägemisala.

Käesolevas töö raames valminud rakenduses on kasutusel variant 1. Valik sai tehtud lihtsmini disainitavate tasemete tõttu.

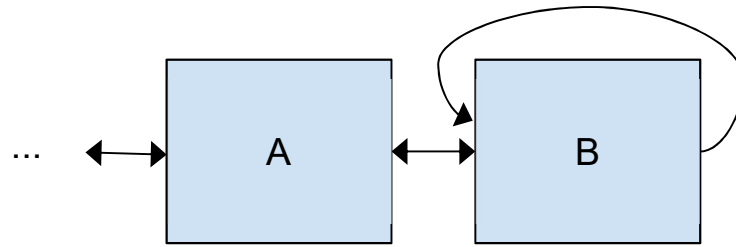
3.3.2 Ühe- ja kahepoolsed ühendused

Vajadus ühepoolsete ühenduste jaoks tekkis soovist teha ühtpidi lõputut koridori (Joonis 15). Sellise konstruktsiooni eesmärk oleks, et mängija saaks ühtpidi lõputult liikuda jäädes korduvas sõlmes liikuma. Samas teisipidi liikudes oleks võimalik saada korduvast sõlmest välja.

Joonisel 15 on sõlmel A kaks kahepoolset ühendust, millest üks on sõlmega B. Sõlmel B on kaks ühendust, millest üks on ühine ühendus sõlmega A ning teine on ühepoolne ühendus iseendaga. Ühendusel iseendaga on määratud selline nihe, et B kuvatakse iseendast paremale. Sõlmede joonistamise suhtes tähendab see seda, et sõlmes B asudes kuvatakse sõlme A sõlmest B vasakul ja sõlme B iseendast paremal. Liikudes sõlmes B paremal asuvasse sõlme B, liigub mängija sinna sõlme ning kuvatakse sõlme B vastavad naabrid, milleks on vasakul pool sõlm A ning paremal pool sõlm B.

Kui mängija liigub sõlmest B paremal pool asuvasse sõlme B, siis eelmine sõlm B kaob ning asemele tekib sõlm A. Seega on selle illusiooni juures oluline, et joonisel 15 sõlmede A ja B parempoolsed servad oleksid sama välimusega, et mängija ei märkaks geomeetria

muutumist. Välimust võivad mõjutada ka sellised asjad nagu teised mängijad või maas olevad liigutatavad objektid.



Joonis 15. Sellel joonisel tähistavad mõlemast otsast noolega ühendused kahepoolseid ühendusi ja ühest otsast noolega tähistatud ühendused ühepoolseid ühendusi.

Eelnevalt kirjeldatud illusiooni on käesolevas töös kasutatud peatükis 4.2.3, et luua lõputut kõrbe.

4. Kasutusnäited

Eelnevad peatükid on kirjeldanud, kuidas sõlmepõhine maailm töötab. Selles peatükis vaatleme kolme erinevat võimalust, kuidas saab sõlmepõhise maailma omadusi kasutada tasemetes disainis:

- otseteed geomeetria vahel;
- korduvad sõlmed;
- ruumiliselt kattuvad alad.

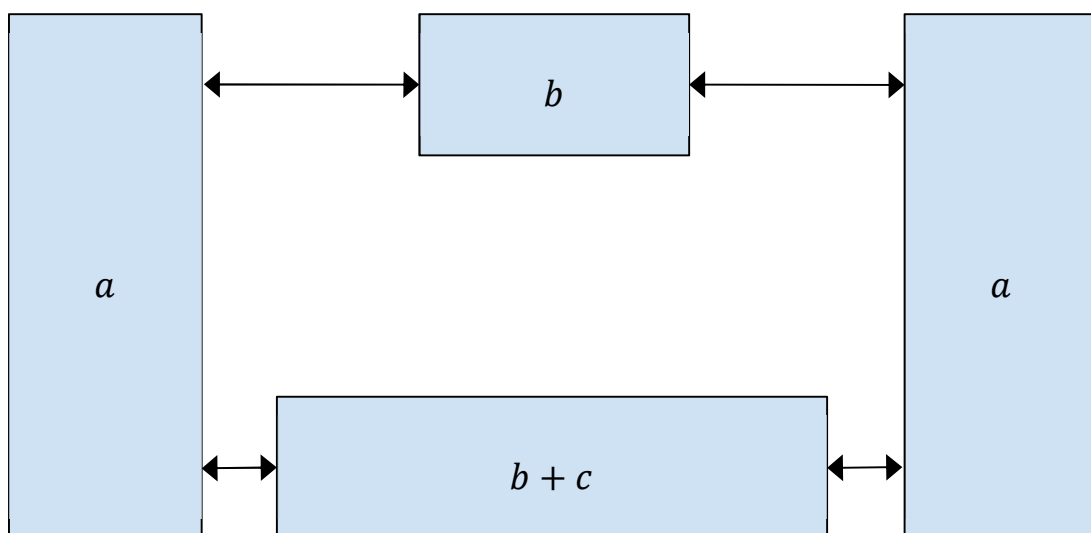
Iga peatükk algab seletusega illusiooni olemuse kohta. Järgnevad kasutusnäidised ning näidiste-sarnased kasutused olemasolevates mängudes. Lõpuks on illusiooni implementatsiooni kirjeldus töö käigus valminud programmis.

4.1 Otseteed geomeetria vahel

Illusiooni eesmärk on tekitada otseid geomeetria vahel, mis tavalise lähenemise puhul poleks võimalikud.

4.1.1 Illusiooni olemus

Lihtsaim näide sellest on neli 90-kraadise pöördega koridori, mis võivad kõik erineva pikkusega olla. Joonisel 16 on konkreetsem näide sellisest konstruktsioonist, kus kaks külgmist koridori on pikkusega a , ülemine koridor on pikkusega b ja alumine koridor on pikkusega $b + c$, kus $a, b, c \in \mathbb{R}^+$. Samas on koridorid teineteise suhtes jätkuvalt 90-kraadise nurga all.



Joonis 16. Tähed sõlmedel tähistavad sõlmede pikemate külgede pikkuseid.

Seega eksisteerib külgmiste koridoride vahel pikk tee pikkusega $b + c$ ning otsetee pikkusega b .

4.1.2 Kasutus

Üks võimalik kasutus oleks mängus, kus on erinevad tasemed ja üks keskne ala. Viimases saab mängija osta varustust ja tasemetes kogutud aardeid müüa. Tihti lahendatakse keskse ala ja tasemete vahel rändamine teleportatsiooniga, mis on üks võimalik süsteem otseteede loomiseks. Näiteks mängus Diablo 2 asusid linnades teleportatsioonialused (illustratsioon 17) ning läbi tasemete asus veel teleportatsioonialuseid. Mängijal oli võimalik kõigi avastatud teleportatsioonialuste vahel rännata. Selline süsteem on mängudes tavaline ning ei paku visuaalselt erilist efekti. Sõlmepõhises maailmas saaks mõelda tasemetest kui pikkadest koridoridest. Keskseks alasse ja tasemetesse saaks luua teleportatsioonialuste asemel ukSED, mis oleksid ühendatud lühikeste koridoridega, läbi mille mängija saab tegelasega kõndida. See jätkaks illusiooni terviklikust, kuid võimatust maailmast.



Illustratsioon 17. Mängus Diablo 2 esinenud Teleportatsioonialus (pildi all osas, keskel)²³.

²³ Diablo II: Lord of Destruction – Mod DB, <http://www.moddb.com/mods/glory-of-nephilim/images/new-act-1-town> (10.05.2017)

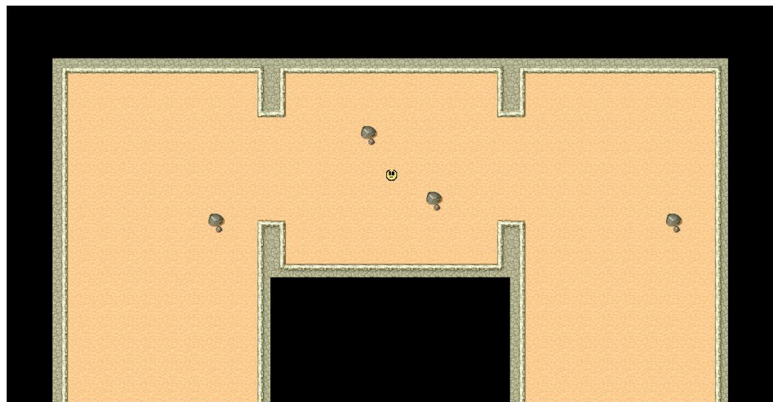
Kui mäng võimaldaks sõlmede vahel dünaamiliselt ühendusi luua ja eemaldada, oleks võimalik lasta mängijal ise valida, kuhu antud otseteekoridorid viivad. See võib välja näha järgnevalt:

1. Mängija leiab tasemest ukse ning tal on varasemast juba mitmeid erinevaid võtmeid.
2. Ta kasutab ukse võtit A ning uks avaneb koridoriga, mis viib võtmele A määratud sihtkohta.
3. Ta mõtleb ümber ja sulgeb ukse ilma sealt läbi astumata.
4. Seejärel avab ta sama ukse kasutades võtit B ning uks avaneb koridoriga, mis viib võtmele B määratud sihtkohta.

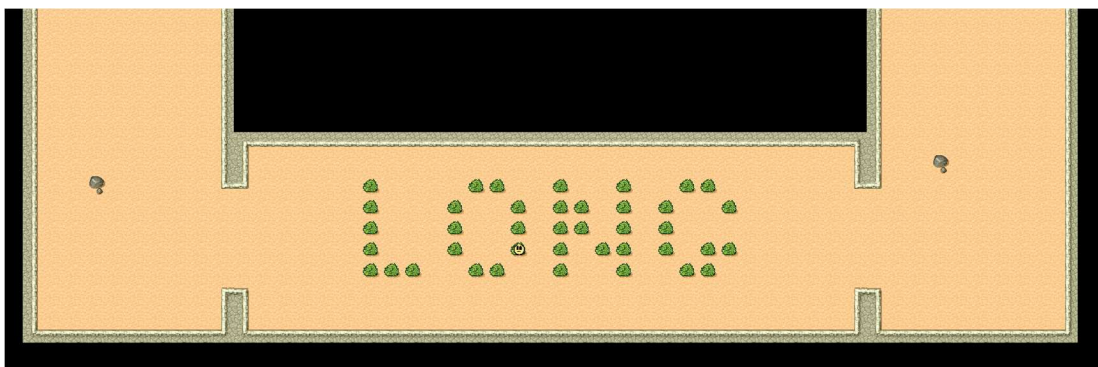
Näiteks saaks mängijatele jagada teatud tegevuste eest võtmeid (taseme lõpetamise eest jne), mis avaks otseteede ukseid kindlatesse kohtadesse.

4.1.3 Implementatsioon

Koos käesoleva tööga valminud programmis on implementeeritud joonisel 16 kujutatud tase (illustratsioonid 18 ja 19). Tegu on nelja sõlmega, mis on omavahel ringjalt ühendatud kahepoolsete ühendustega.



Illustratsioon 18. Lühike koridor.



Illustratsioon 19. Pikk koridor.

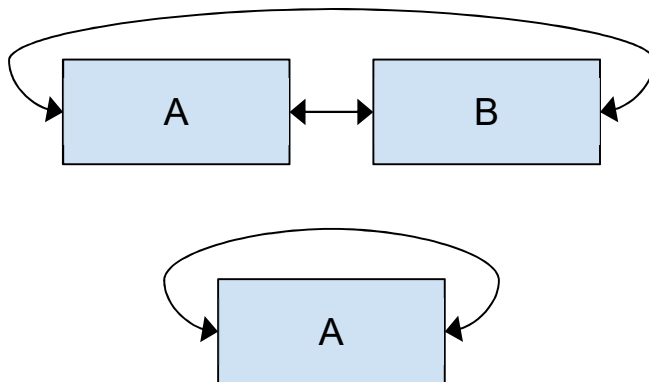
Visuaalselt jätavad koridorid mängijale tervikliku mulje ning on näha, et koridorid on täisnurga all teineteise suhtes. Samas illustratsioonil 18 asuv keskmine sõlm ja illustratsioonil 19 asuv keskmine sõlm on erinevate pikkustega.

4.2 Lõpmatu koridor

Lõpmatu koridori eesmärk arvutimängudes on luua mängijale olukord, kus tal tekib illusioon edasi liikumisest, kuid ta ei jõua otseselt kuskile uude kohta. Antud töös kirjeldatakse ühepoolseid ja kahepoolseid lõpmatuid koridore. Ühepoolses avastatakse olukord mängija poolt tihti siis, kui ta lõpmatus koridoris otsustab tagasi liikuda. Kahepoolses on tal vajalik märgata, et edasi liikudes jõuab ta samasse kohta, kus ta juba varem oli. Tagasi liikumine kahepoolses lõpmatus koridoris on samuti lõpmatu.

4.2.1 Illusiooni olemus

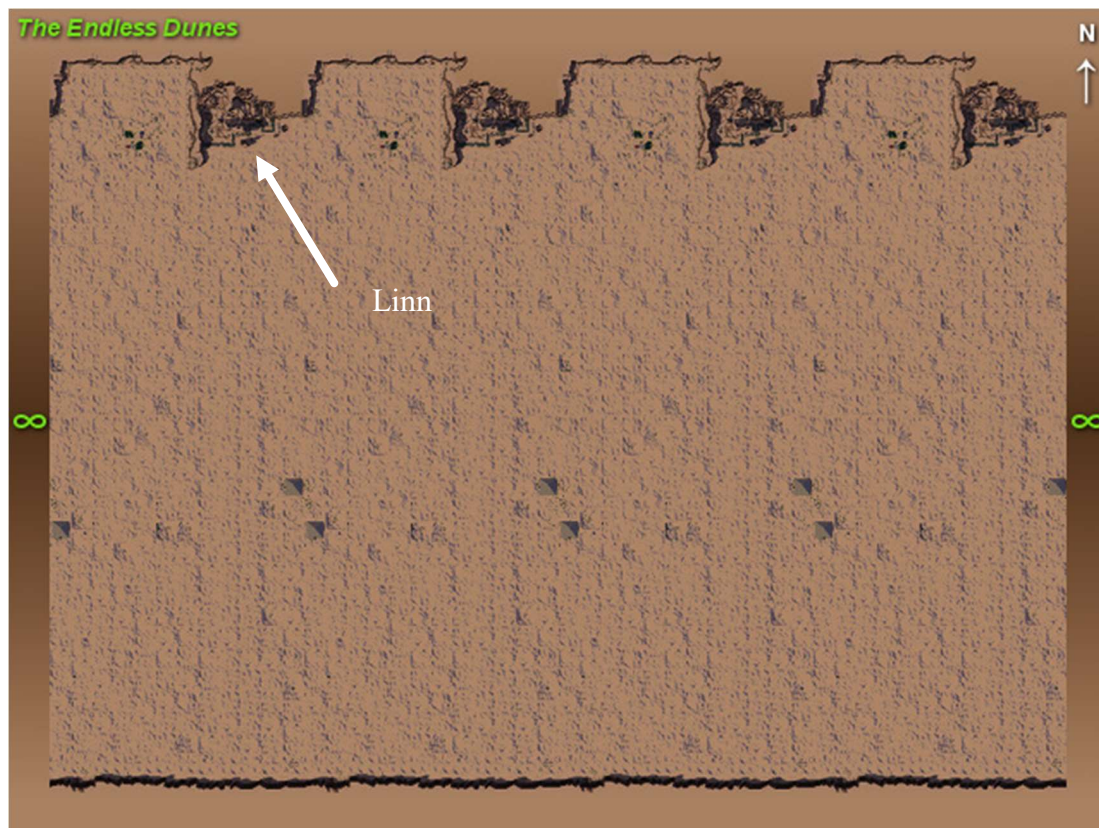
Graafiteooria kohaselt on tegu tsükliga. Tsüklikks nimetame ahelat, kus algustipp ühtib lõputipuga (joonis 20, ülemine). Ahela saab moodustada ka ainult üks serv – nt silmus [16]. Silmus on ühest servast koosnev tsükkel (joonis 20, alumine).



Joonis 20. Lihtsad näited tsüklitest.

4.2.2 Kasutus

Mängus Dungeon Siege (2002) on ala nimega “The Endless Dunes” (“Lõputud Düünid”, illustratsioon 21). Sellel alal ida või lääne suunas liikudes on võimalik lõputult liikuma jääda kuna ala moodustab lõputult korduva koridori. Põhja suunas oli läbi linna sissepääs alale ning lõunas oli ees läbimatu sein.



Illustratsioon 21. “Lõputud Düünid” mängust Dungeon Siege neljakordselt kuvatuna²⁴.

Teatud kitsendustega arvestades (lähemalt peatükis 4.2.3) on võimalik luua ka kolmes suunas lõputu ala, kasutades selleks ühepoolseid ühendusi. Sisuliselt oleks üks keskne ala, kuhu on võimalik ühelt poolt siseneda. Keskelt alalt üheski teises suunas peale sissepääsu poole liikudes jääb mängija endale märkamatu ühepoolses lõpmatus koridoris ringlema. Illusiooni eesmärk on jätta mängijale mulje, et ta liigub tegelasega lähtekohast üha kaugemale, kuigi tegelikkuses käib ta sisuliselt kohapeal ringi. Kui “Lõputute Düünide” puhul linnast näiteks läänesuunas liikudes saabus mängija linna taas idasuunast, siis ühepoolsete ühenduste puhul seda ei juhtuks.

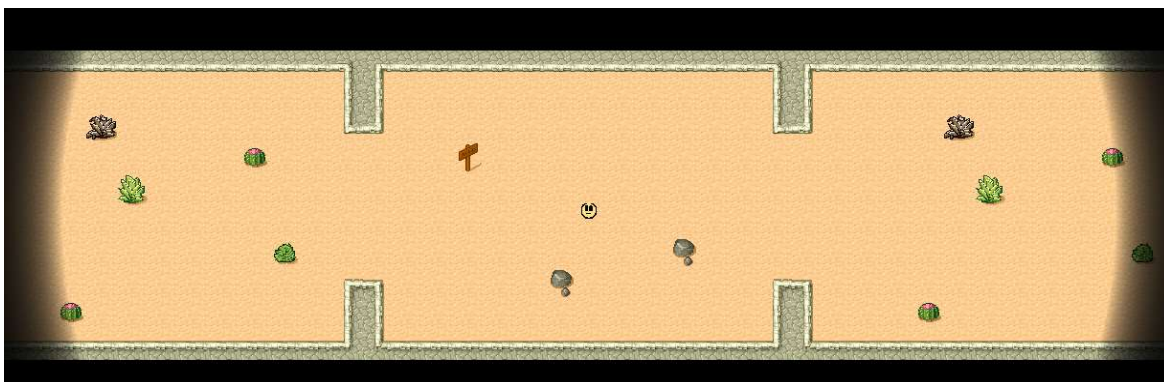
Tavaliselt kasutatakse mängumaailma piirideks läbimatuid seinu. Selle illusiooni võimalikuks kasutuseks oleks luua mängumaailma ümber nähtamatu piire.

4.2.3 Implementatsioon

Koos käesoleva tööga valminud programmis on implementeeritud mõlemad joonisel 20 kujutatud lõputud koridorid ja kolmelt küljelt lõputu ala.

²⁴ Endless Dunes – Dungeon Siege Wiki, http://dungeonsiege.wikia.com/wiki/Endless_Dunes (10.05.2017)

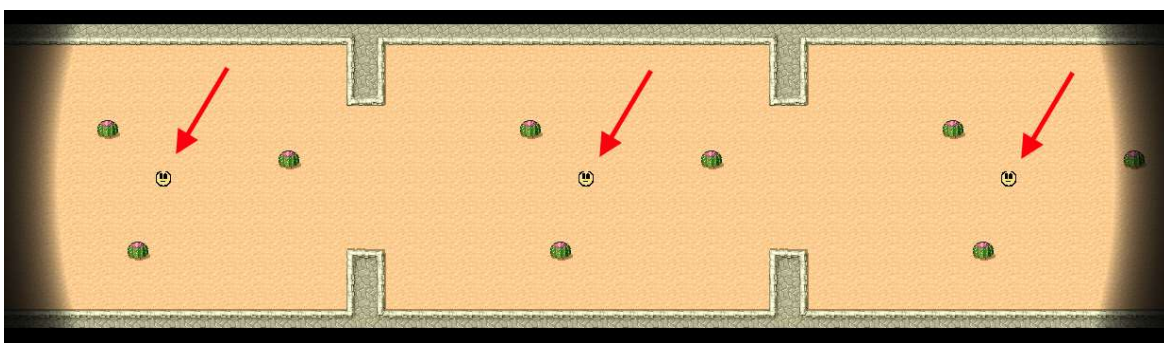
Joonisel 20 kujutatud koridorid on äärmiselt lihtsad implementeerida. Kahe sõlmega näidises on sõlmed A ja B ning kaks ühendust nende vahel on sõlmest A sõlme B ja sõlmest B sõlme A. Programmis implementeeritud variant on näha illustratsioonis 21.



Illustratsioon 21. Kahe erineva sõlmega koridor.

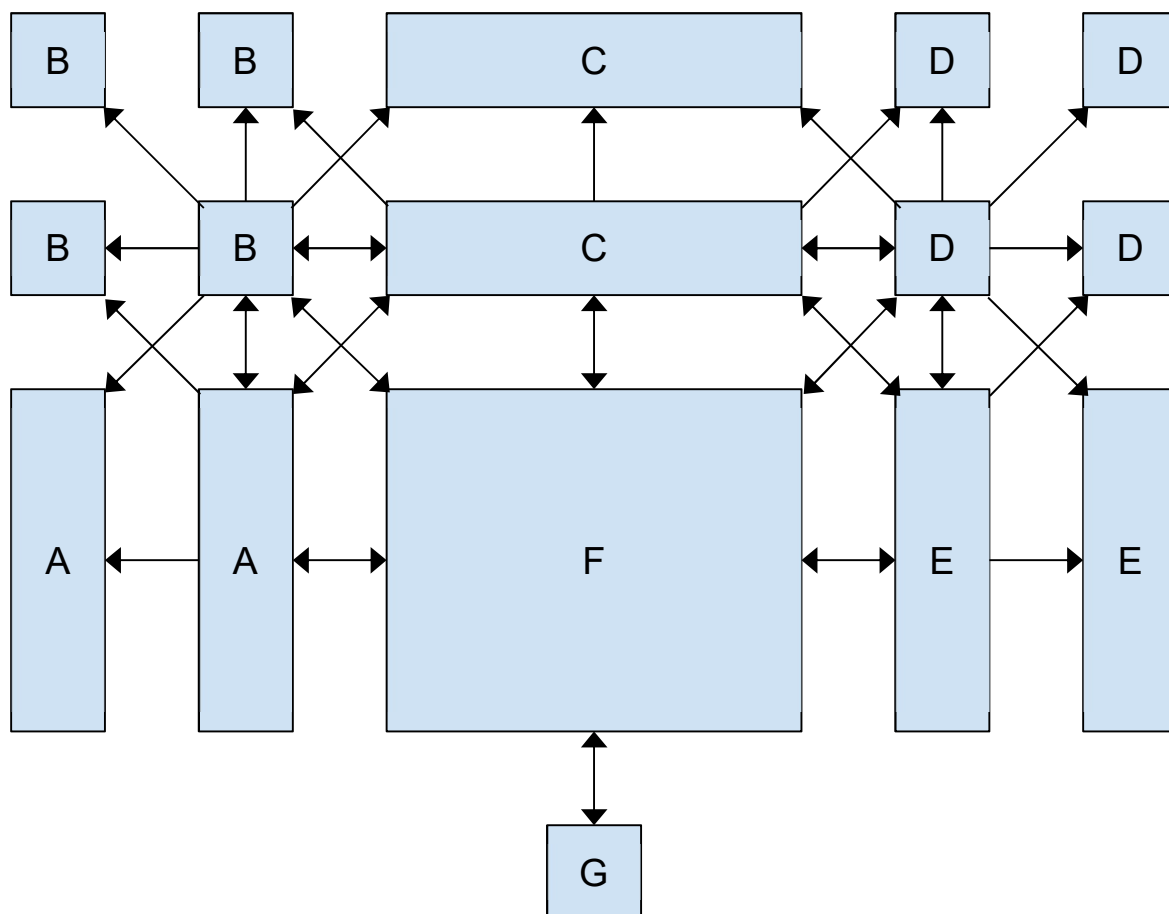
Vasakul ja paremal ääres asuvad sõlmed on sama sõlm.

Ühe sõlmega näidis joonisel 20 on veelgi lihtsama implementatsiooniga. Sisuliselt on programmis defineeritud kõigest üks sõlm ning ühendus iseendaga. Illustratsioonilt 22 on aga näha veel üks iseärasus. Kuna pildil on sama sõlm kolmekordselt ning mängija tegelane on sisuliselt objekt sõlmes, siis kuvatakse teda ka igas sõlmes, mis ekraanile kuvatakse. Seega kuvatakse tegelast kolmekordselt.



Illustratsioon 22. Ühe sõlmega koridor. Teglane on märgitud punase noolega.

Kolmest küljest lõpmatu ala oli kõige keerulisem programmis implementeerida. Sõlmed ja nende vahelised ühendused on toodud välja joonisel 23. Unikaalsed sõlmed on A, B, C, D, E, F ja G. Sõlmed A, B, C, D ja E on joonisel kujutatud mitmekordselt, et paremini näidata ühendusi sõlmede vahel.



Joonis 23. Kolmest suunast lõputu ala.

Sõlm G on sissepääs sõlme F. Sõlmed A, F ja E on alumisest küljest piiratud seinaga, peale sõlmede F ja G ühise osa. Ülejäänud sõlmed pole takistustega piiratud. Oluline on, et üheski sõlmes asudes ei näeks seletamatuid katkendeid geomeetrias.

Sõlmed A, C ja E on sisuliselt ühepoolsed lõpmatud koridorid. Kuid ainuüksi nendest sõlmedest ei piisa, et maailm terviklik tunduks. Näiteks sõlmel C vasakus suunas ja sõlmel A üleval suunas oleks tühi ala. Selle lahenduseks on seal sõlm B, mis on omakorda ühepoolne lõpmatu koridor, kuid seda nii vasakult, ülevalt ja vasakult-ülevalt diagonaalsest suunast. Sedasi paistab maailm terviklik ning üheski suunas liikudes ei asu mängija tegelane sõlmest F kunagi kaugel.

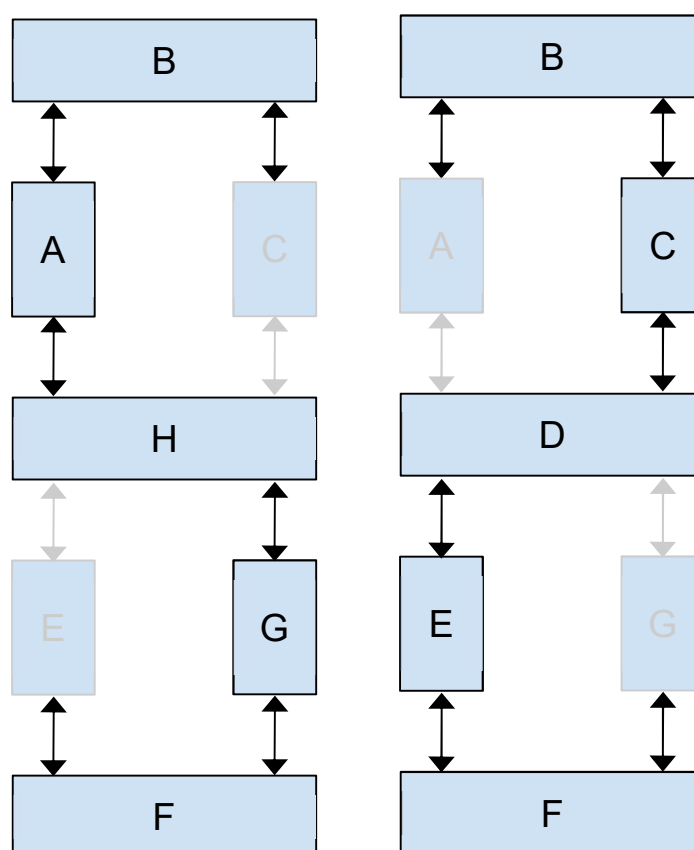
4.3 Ruumiliselt kattuvate aladega geomeetria

Selle peatüki eesmärk on uurida artiklis “The Continuous World of Dungeon Siege” [2] olevat lauset “*The world can, and often does, intersect itself, and end up bent like a pretzel. Two regions can completely overlap, where one is faded out and the other is visible.*” Eesti

keeles oleks see “Maailm võib, ja tihti ka teeb sedasi, lõikuda iseendaga ning olla väänatud nagu kringel. Kaks ala võivad täielikult kattuda, kus üht ei kuvata ning teist kuvatakse.”

4.3.1 Illusiooni olemus

Olgu maailmas sõlmed tähistustega A, B, C, D, E, F, G ja H, mis on omavahel tsükliliselt ühendatud (joonis 24). Ruumiliselt paiknevad sõlmed D ja H samas kohas. Kuna alati on kuvatud ainult aktiivne sõlm ja tema naabrid ning sõlmedel D ja H ühiseid naabreid pole, siis neid ei kuvata kunagi ka samal ajal.



Joonis 24. 8-paigutuses olev ruum. Vasakul on toodud esile sõlmega H seotud sõlmed ning paremal sõlmega D seotud sõlmed.

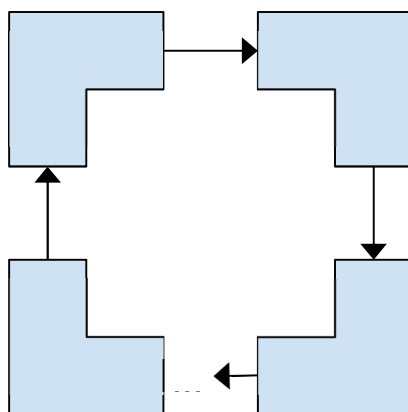
Sedasi on võimalik teha ruumiliselt kattuvaid sõlmi. Järgnevalt vaatleme ka sellise konstruktsiooni võimalikke kasutusi.

4.3.2 Kasutus

Ruumis kattuvuse omaduse teeb võimalikuks luua maailmu, mis võivad iseendaga lõikuda. See annab vabamad käed tasemete disainimisel, sest peab vähem arvestama tasemete paigutusega. Lisaks on sellel huvitavaid kasutusvõimalusi, millest on järgnevalt lähemalt kirjutatud.

4.3.2.1 Näide mängust P.T.

Järgnev näide tuleb mängus P.T. (*Playable Teaser*)²⁵, kus mängija läbis näiliselt sama koridori korduvalt, aga iga kord oli keskkonnas midagi muutunud. Selles mängus olid 90-kraadise pöördega koridorid (joonis 25). Iga koridor algas uksega ja lõppes väikese allatrepiga, mille lõpus oli uks. Koridorid moodustasid sisuliselt allapoole mineva trepi. Kuna trepp oli pimedas ning ainult mõned astmed pikk, võis mängija seda mitte märgata. Sedasi võis jääda mulje, et ruumid kattuvad teineteisega ning on tegu võimatu maailmageomeetria. Sõlme põhises maailmas oleks võimalik luua P.T. maailmaga sarnast maailma, aga ilma allapoole minevate treppideta ning ka tegelikult ruumiliselt kattuvate koridoridega.

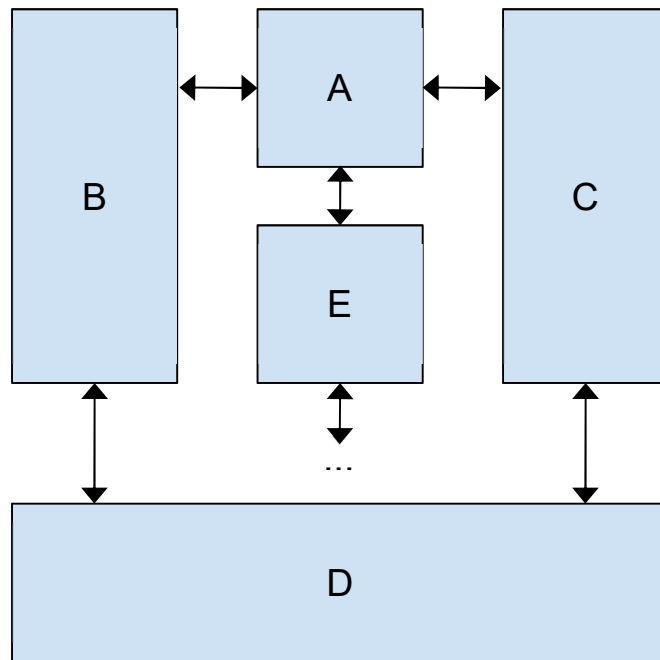


Joonis 25. Ruumide paigutus mängus P.T.

4.3.2.2 Väljast väike, seest suur maja

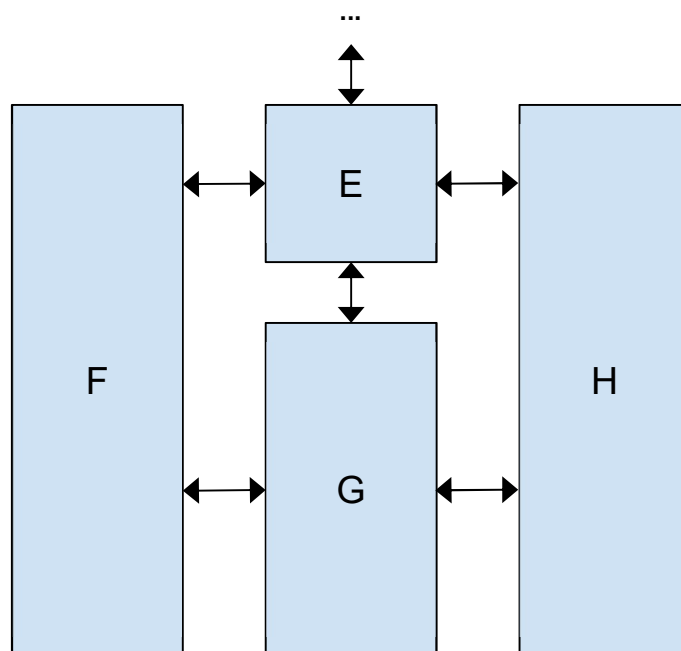
Veel üks kasutus ruumiliselt kattuvatele sõlmedele oleks teha näiteks väljast väike maja, millesse sisse astudes avaneks aga suur maja. Joonisel 26 on kujutatud üks viis, kuidas saaksid sõlmed asuda, et sellist illusiooni luua. Sõlmed A, B, C ja D on majast väljas asuvad sõlmed. Nad moodustavad maailmageomeetria maja ümber. Sõlm E oleks nõ esik, mis asuks maja suure sisemuse ja välismaailma vahel. Sõlmedes B, C ja D olles näeks mängija ainult maja välisseinu. Esik tekiks mängijale nähtavale alles siis kui ta asub tegelasega sõlmes A.

²⁵ P.T. (video game) – Wikipedia, [https://en.wikipedia.org/wiki/P.T._\(video_game\)](https://en.wikipedia.org/wiki/P.T._(video_game)) (10.05.2017)



Joonis 26. Majaväline geomeetria.

Astudes sõlmest A sõlme E kuvatakse ülejäänud majasisene geomeetria (joonis 27) ning jääb nähtavale majaesine sõlm A. Astudes edasi majja sisse, kaob ka sõlm A ning mängija asubki täielikult maja sees. Joonisel 27 kujutatud majas sees asuvad sõlmed F, G ja H võivad olla kui tahes suured, sest maja välissõlmi enam ei kuvata ning ei tekiks visuaalset ülekattet majaväliste sõlmedega.



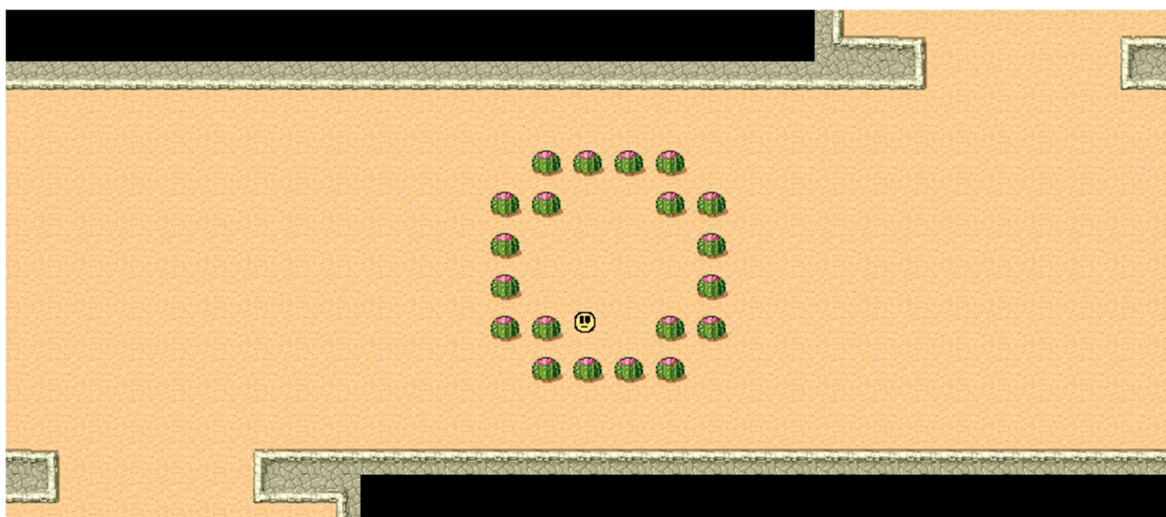
Joonis 27. Majasisene geomeetria.

4.3.3 Implementatsioon

Antud töös on implementeeritud joonisel 24 kujutatud 8-paigutuses ruum sedasi nagu on kirjeldatud peatükis 4.3.1. Illustratsioonil 28 on näha sõlm H jooniselt 24. Illustratsioonil 29 on näha sõlm D jooniselt 24. Sõlmed H ja D kattuvad ruumiliselt, kuid kuna nendesse viivad erinevad teed, siis alati kuvatakse neist ainult ühte korraga.



Illustratsioon 28. Sõlm H jooniselt 24 – implementeeritult valminud programmis.



Illustratsioon 29. Sõlm D jooniselt 24 – implementeeritult valminud programmis.

5. Kokkuvõte

Käesolevas töös kirjeldati võimalikke geomeetriaid ja meetodit nende implementeerimiseks kasutades sõlmepõhist maailma. Töö kirjeldas selle ülesande lahendamisel tekkinud probleeme ja võimalikke lahendusi. Näiteks mängija vaatevälja piiramine, et varjata sõlmedevahelisi katkendeid ning probleemi lahendusena implementeeritud valgussõõr.

Antud lähenemist võimalike geomeetria kujutamiseks illustreerib töö raames valminud rakendus. Implementeeritud on 5 erinevat taset, mis näidistavad 3 erinevat tüüpi võimalikku geomeetria. Need kolm erinevat tüüpi on otseteed geomeetria vahel, lõpmatud koridorid ja ruumiliselt kattuvate aladega geomeetria.

Töö teema on olnud minu jaoks väga põnev ja arendav. Plaanin selle teemaga tegelemist jätkata ja mõelda välja veel erinevaid võimalikke geomeetriaid või kasutusi, mida sõlmepõhine maailm võimaldab. Lisaks on võimalikud tulevikuplaanid arendada programmi edasi mänguks ning implementeerida süsteem kolmemõõtmelises mängumaailmas. Loodan, et töö pakub inspiratsiooni, süstemaatilist käsitlust ja juhendust teistele teemast huvitujatele.

6. Viidatud kirjandus

- [1] Penrose LS, Penrose R. Impossible Objects: A Special Type of Visual Illusion. The British Journal of Psychology. 1958 veebruar;49(1):3. Saadaval: <http://onlinelibrary.wiley.com/doi/10.1111/j.2044-8295.1958.tb00634.x/full>
- [2] Bilas S. The Continuous World of Dungeon Siege. Game Developers Conference GDC 2003; 2003; San Jose, CA, USA. 16 p. Saadaval: http://scottbilas.com/files/2003/gdc_san_jose/continuous_world_paper.pdf
- [3] Wikipedia: the free encyclopedia [Internet]. St. Petersburg (FL): Wikimedia Foundation, Inc. 2001. Impossible object [muudetud 2017 märts 06; tsiteeritud 2017 mai 6]; [umbes 6 ekraani]. Saadaval: https://en.wikipedia.org/wiki/Optical_illusion
- [4] Freud E, Avidan G, Ganel T. Holistic processing of impossible objects: Evidence from Garner's speeded-classification task. Vision Research [Internet]. 2013 detsember 13 [tsiteeritud 2017 mai 06]; 93:10-18. Saadaval: <http://www.sciencedirect.com/science/article/pii/S0042698913002368> DOI: <https://doi.org/10.1016/j.vis-res.2013.10.001>
- [5] Wikipedia: the free encyclopedia [Internet]. St. Petersburg (FL): Wikimedia Foundation, Inc. 2001. Optical illusion [muudetud 2017 aprill 17; tsiteeritud 2017 mai 06]; [umbes 3 ekraani]. Saadaval: https://en.wikipedia.org/wiki/Impossible_object
- [6] Seckel A. Masters of Deception: Escher, Dalí & the Artists of Optical Illusion. New York: Sterling Publishing Company; 2004. 320 p.
- [7] TotallyHISTORY [Internet]. The Waterfall. Saadaval: <http://totallyhistory.com/the-waterfall/>
- [8] TotallyHISTORY [Internet]. Ascending and Descending. Saadaval: <http://totallyhistory.com/ascending-and-descending/>
- [9] PlayStation. echochrome™ Official Trailer [Internet]. 2008 oktoober 03 [tsiteeritud 2017 mai 09]. Video: 1 min 38 sek. Saadaval: <https://www.youtube.com/watch?v=GybxIwfU4rI>
- [10] Truu A. PROGRAMMEERIMISE ALUSED. [Internet]. Tartu: Tartu Ülikooli Teaduskool; 2007 [tsiteeritud 2017 mai 08]. Peatükk 8, Mittelineaarsed andmestruktuurid. Saadaval: <http://www.ttkool.ut.ee/comp/kaug/prog1/prog009.html>

- [11] Bravo EG, García JA. Yturalde: impossible figure generator. SIGGRAPH '15 ACM SIGGRAPH Art Papers [Internet]. 2015 juuli 31 [tsiteeritud 2017 mai 08]; 366-374. Saadaval: <http://dl.acm.org/citation.cfm?id=2810183>
DOI: <https://doi.org/10.1145/2810177.2810183>
- [12] Nakatsu K, Takahashi T, Moriya T. Animating impossible figure from twisted torus to penrose stairs. SA '13 SIGGRAPH Asia 2013 Posters [Internet]. 2013 [tsiteeritud 2017 mai 08]; Article No. 33. Saadaval: <http://dl.acm.org/citation.cfm?id=2542341>
- [13] Sedgewick R, Wayne K. Introduction to Programming in Java [Internet]. New Jersey: Princeton University; [kuupäev pole teada]; [viimati muudetud 2014 aprill 22; tsiteeritud 2017 mai 08]. Peatükk 9.1, Floating Point. Saadaval: <http://introcs.cs.princeton.edu/java/91float/>
- [14] Smith R. The NVIDIA GeForce GTX 1080 & GTX 1070 Founders Editions Review: Kicking Off the FinFET Generation. AnandTech. [Internet]. 2016 juuli 20. [tsiteeritud 2017 mai 08]. Saadaval: <http://www.anandtech.com/show/10325/the-nvidia-geforce-gtx-1080-and-1070-founders-edition-review/28>
- [15] kappaOne. Java OpenGL Math Library (JOML). 2015 juuni 01 [tsiteeritud 2017 mai 08]. In: Lightweight Java Game Library [Internet]. Saadaval: <https://blog.lwjgl.org/maths-library/>
- [16] Koit M. GRAAFITEOORIA PÕHIMÕISTED. [Internet]. Tartu: Tartu Ülikooli Teaduskool; [kuupäev pole teada] [tsiteeritud 2017 mai 08]. Saadaval: http://www.teaduskool.ut.ee/sites/default/files/teaduskool/oppetoo/mat_gymn_graafiteooria.pdf

Lisad

I. Programm

Programm on tööga kaasas olevas kaustas “Programm”. Käivitamiseks peab olema arvutisse paigaldatud Java 8 JDK ning rakenduse juurkaustas tuleb jooksutada käsurea käsku „gradlew run“.

Nõuded arvutile:

- Operatsioonisüsteem: Windows 7 või uuem
- Protsessor: 2 GHz
- Muutmälu (RAM): 512MB
- Kõvaketta ruum: 10MB
- Videokaart: OpenGL 3.0 või uuema tugi
- Internetiühendus

Lähtekood on saadaval Bitbucket repositooriumis: <https://bitbucket.org/ivov/thesisproject/>

Praktilise osa valmides tuli välja, et kui rakendus ehitada ja pakendada lihtsasti käivitatavaks programmiks, on libtiled-java teegil probleeme kaardifailides kasutatud paanikomplektifailide sisselaadimisega. Probleem jäi lahendamata ajaliste piirangute tõttu. Võimalikuks lahenduseks oleks pakendada kaardifailid JAR-failist eraldi. Hetkel tuleb aga programmi käivitamiseks ehitada lähtekood käsurea käsuga kokku ning see ise käivitada. Seetõttu on vajalik ka Java JDK versioon, mis on mõeldud Java programmide koostamiseks.

II. Programmi kasutusõpetus

Klahvid programmi kasutamiseks on kirjeldatud allolevas tabelis 28:

Klahv		Tegevus
W	↑	Ülesse liikumine
A	←	Vasakule liikumine
S	↓	Alla liikumine
D	→	Paremale liikumine
I		Valgussõõri suurendamine
K		Valgussõõri kitsendamine
O		Valgussõõri vaikumisi suuruse taastamine
1		Kahe sõlmega lõpmatu koridor
2		Erinevate pikkustega koridorid
3		8-paigutuses ruum
4		Ühe sõlmega lõpmatu koridor
5		Kolmest suunast lõpmatu kõrb

Tabel 28. Klahvid programmi kasutamiseks.

III. Videod

Tööga on kaasas videod (kaustas “Videod”), mis demonstreerivad rakenduses implementeeritud näiteid:

- kaks_solme.mp4 – Joonisel 20 kujutatud kahe sõlmega lõpmatu koridor.
- pikk_koridor.mp4 – Joonisel 16 kujutatud ruum.
- 8_paigutus.mp4 – Joonisel 24 kujutatud 8-paigutuses ruum.
- yks_solm.mp4 – Joonisel 20 kujutatud ühe sõlmega lõpmatu koridor.
- korb.mp4 – Joonisel 23 kujutatud kolmest küljest lõpmatu ala.

IV. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Ivo Voika**,
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Võimatu geomeetria sõlmepõhises maailmas,
(*lõputöö pealkiri*)

mille juhendaja on Raimond-Hendrik Tunnel,
(*juhendaja nimi*)

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **11.05.2017**